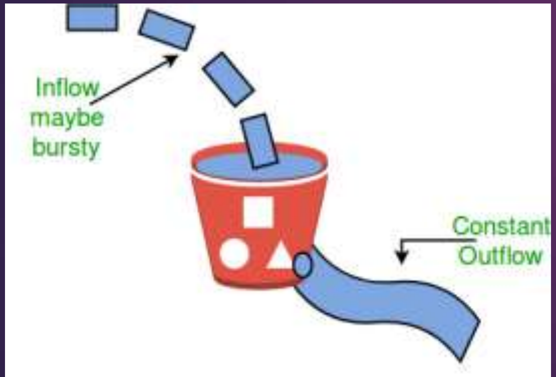
CUBIC-FIT: A High Performance Congestion Control Algorithm

CSE 322: Computer Network Sessional (NS2)

PRESENTED BY,

AYESHA BINTE MOSTOFA

STUDENT ID: 1805062



Reference

Paper Link :

https://www.researchgate.net/publication/260533734_CUBICFIT_A_high_performance_and_tcp_CUBIC_friendly_congestion_control_algorithm

Author:

Jingyuan Wang, Jiangtao Wen, Yuxing Han, Jun Zhang, Chao Li and Zhang Xiong

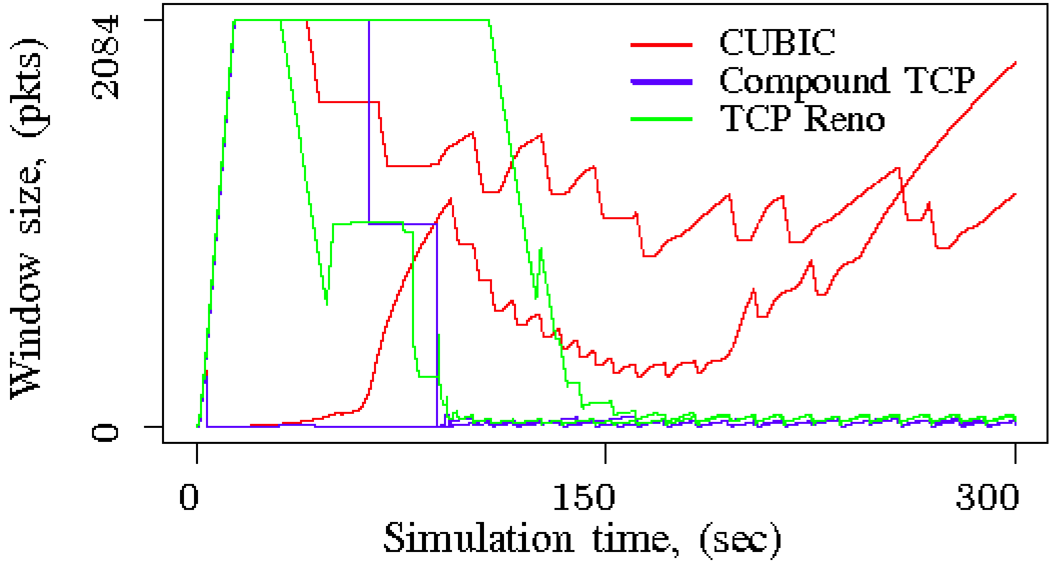
- Published In: IEEE Communications Letters (Volume: 17, Issue: 8, August 2013)

Motivation

Increase of Linux based networks

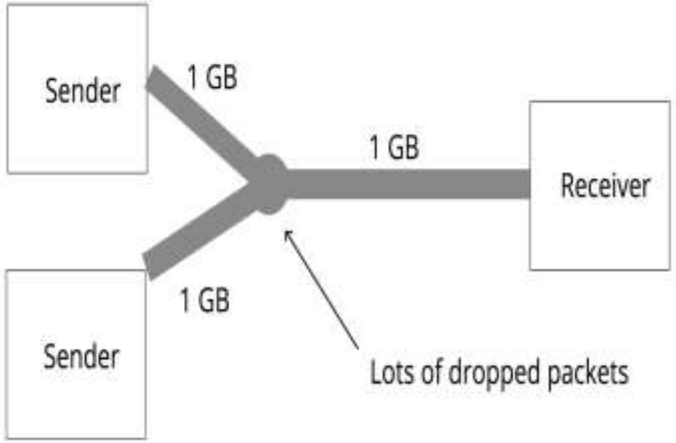
Linux Kernel uses TCP Cubic

Cubic friendly Algorithm is required like existing TCP Reno friendly Algorithms.



Existing Algorithm : CUBIC

2 challenges for the CUBIC algorithm and CUBIC-dominated networks.



I. ***Performance over wireless links***. CUBIC is designed for large BDP networks but **not wireless networks**. CUBIC uses packet losses as network congestion symptoms, so random physical and MAC layer artifacts introduced packet losses can cause performance degradation of TCP CUBIC.

II. ***CUBIC friendliness***. TCP flows in CUBIC-dominated networks should stay friendly with other CUBIC flows. However, most existing algorithms are designed to Reno-dominated networks and maintain friendliness with Reno but not CUBIC, which introduces serious **fairness problem**.

$$T_0 - T' = \frac{1}{D}\sqrt[4]{\frac{1.9D^3}{P^3}} - \frac{1}{D+q'}\sqrt[4]{1.9\left(\frac{D+q'}{P+p'}\right)^3}.$$

Improved Algorithm: CUBIC-FIT

Delay-based TCP to extend the CUBIC algorithm framework

Simulation of the behavior of multiple CUBIC flows **in a single TCP connection** to fully utilize network capacity.

End-to-end queuing delay to adjust the simulated CUBIC flow number.

Cubic Vs Cubic-Fit

- When there are no packet losses, TCP CUBIC network capacity :

$$w_{cubic} = C(t-I)^3 + w_{max}$$

$$where\ , I = \sqrt[3]{(w_{max} b)/C}$$

- Transmission rate of a CUBIC flow,

$$T_{cubic} = \frac{1}{RTT} \sqrt[4]{\frac{C(4-b)}{4b} (\frac{RTT}{PLR})^3}$$

- When there are no packet losses, TCP CUBIC -Fit network capacity :

$$w_{fit\ cubic} = .4(Nt-I)^3 + w_{max}$$

$$where\ , I = \sqrt[3]{(10\ w_{max})/(19N+1)}$$

- Transmission rate of a CUBIC-Fit flow

$$T_{fit\ cubic} = N.T_{plain\ cubic} = \frac{N}{RTT} \sqrt[4]{\frac{C(4-b)}{4b} (\frac{RTT}{PLR})^3}$$

where PLR = average packet loss rate, PLR , and RTT = average round-trip time RTT

Cubic-Fit Improvement

The behaviors of N plain CUBIC flows can be emulated in a single connection. N is an important parameter to control aggression in CUBIC-FIT. N values that are too small may cause network capacity under-utilization, but N values that are too large may cause starvation in other TCP flows.

$$So,\ N_{t+1} \coloneqq \max\left\{1\ , N_t\ +\ 1 - N_t\ \frac{RTT_t\ - RTT_{min}}{aRTT}\right\}$$

$$T_{fit\ cubic}\ = \max\left\{\frac{1}{RTT}, \frac{a}{q}\right\}\ \sqrt[4]{1.9(\frac{RTT}{PLR})^3}$$

$$where, q\ = RTT_t\ - RTT_{min}\ ,\ and\ a\ \coloneqq \min\{\frac{1}{10}\ , \frac{RTT_{max} - RTT_{min}}{2RTT_{max}}\}$$

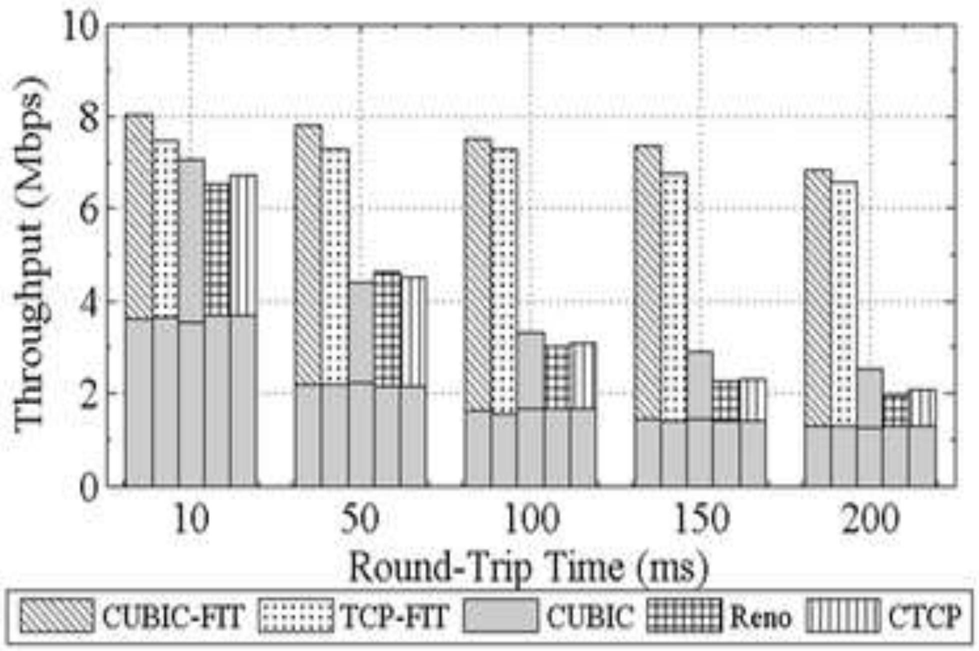
From the equation of N and Throughput, we get it , a low queuing delay q leads to an increase in N to improve performance if networks are not fully utilized. Conversely, CUBIC-FIT decreases N to maintain friendliness with plain CUBIC flows when network suffers long queuing delays informed congestion.

Improvement

Throughput performance improvement over wireless networks more than plain CUBIC.

Performance over large range of network

Maintain graceful friendliness with widely deployed plain CUBIC servers.



Thank You!